

Débruiteur “descente de gradient” pour la convergence d’une méthode Plug-and-Play

Samuel HURAUULT¹, Arthur LECLAIRE¹, Nicolas PAPADAKIS¹

¹Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251,F-33400 Talence, France

samuel.hurault@math.u-bordeaux.fr

Résumé – Les méthodes Plug-and-Play (PnP) forment une classe d’algorithmes itératifs pour la résolution de problèmes inverses en imagerie où la régularisation est effectuée par un débruiteur générique. Bien que produisant de très bons résultats visuels pour différents problèmes d’imagerie, les preuves de convergences actuelles reposent sur des hypothèses peu réalistes pour le débruiteur ou sur la forte convexité de l’attache aux données. Nous proposons un nouveau type de méthode PnP exploitant un débruiteur qui s’écrit comme une étape de descente de gradient sur une fonctionnelle paramétrée par un réseau de neurones profond. Nous démontrons que ce nouvel algorithme converge vers un point stationnaire d’une fonctionnelle explicite. En appliquant cet algorithme au problème de défloutage, nous confirmons empiriquement les résultats théoriques de convergence, ainsi que la haute qualité du résultat de la restauration, à l’état de l’art tant quantitativement que qualitativement.

Abstract – Plug-and-Play (PnP) methods constitute a class of iterative algorithms for imaging problems where regularization is performed by an off-the-shelf denoiser. Although PnP methods can lead to tremendous visual performance for various image problems, the few existing convergence guarantees are based on unrealistic hypotheses on the denoiser, or limited to strongly convex data-fidelity terms. We propose a new type of PnP method for which the denoiser is realized as a gradient descent step on a functional parameterized by a deep neural network. We show that the proposed PnP algorithm is a convergent iterative scheme that targets stationary points of an explicit functional. We apply our algorithm to the deblurring inverse problem. These experiments confirm the convergence results and show that this new algorithm reaches state-of-the-art restoration performance, both quantitatively and qualitatively.

1 Introduction

Les problèmes de restauration d’image sont généralement formulés comme des problèmes inverses de la forme

$$x^* \in \arg \min_x f(x) + \lambda g(x) \quad (1)$$

où f mesure l’attache à une donnée dégradée y , et g est un terme de régularisation pondéré par $\lambda \geq 0$. Généralement, la dégradation d’une image propre \hat{x} peut-être modélisée par un opérateur linéaire $y = A\hat{x} + \xi$, où A est une matrice de dégradation et ξ un bruit blanc Gaussien. Pour la recherche du maximum a posteriori (MAP) l’attache aux données est liée à la vraisemblance $f(x) = -\log p(y|x) = \frac{1}{2\sigma^2} \|Ax - y\|^2$, alors que le terme de régularisation est lié au choix d’un a priori.

Étant donné que le problème de restauration est mal posé, l’apport de connaissance a priori par le terme de régularisation est crucial. De nombreux travaux ont porté sur la difficile tâche de la conception d’un a priori efficace, en utilisant par exemple la variation totale [14], la parcimonie dans une base [10], ou en l’apprenant directement d’une base d’images propres [9].

Le problème (1) n’a en général pas de solution exacte et un algorithme d’optimisation est nécessaire pour le résoudre. Les algorithmes proximaux [4] du premier ordre (comme le "half-quadratic splitting" (HQS) [5]) opèrent séparément sur f et g via l’opérateur proximal

$$\text{Prox}_f(x) = \arg \min_z \frac{1}{2} \|x - z\|^2 + f(z). \quad (2)$$

Les méthodes "Plug-and-Play" (PnP) [17] sont construites à partir des algorithmes proximaux en remplaçant Prox_g par un débruiteur générique, par exemple un réseau de neurones profond pré-entraîné. Ces méthodes permettent d’obtenir des résultats à l’état de l’art [18, 19] pour différents problèmes de restauration d’image. Cependant, puisqu’un débruiteur générique ne peut généralement pas être exprimé comme un opérateur proximal [12], les résultats de convergence des algorithmes proximaux ne sont pas directement applicables aux algorithmes PnP. De nombreux travaux analysent alors la convergence de ces schémas en se limitant à des attaches aux données fortement convexes [15], ce qui exclut de nombreux problèmes de restauration, ou en proposant l’utilisation de débruiteurs non-expansifs [16], ce qui peut dégrader sérieusement leurs performances [7]. De plus, la régularisation g est rendue implicite par l’opération de débruitage. Les algorithmes PnP ne cherchent donc pas à minimiser une fonctionnelle explicite, ce qui limite leur interprétation et leur contrôle numérique.

Contributions. Nous proposons ici un schéma PnP avec de nouvelles garanties théoriques de convergence et des résultats de restauration d’image au niveau de l’état de l’art. À partir de l’algorithme PnP-HQS, nous contraignons le débruiteur à être un champ de gradients, sans compromettre ses performances. Nous garantissons la convergence de ce nouvel algorithme vers un point stationnaire d’une fonctionnelle explicite.

2 Plug-and-Play descente de gradient

La méthode proposée est basée sur la version PnP de half-quadratic-splitting (PnP-HQS) [19] qui remplace l'opérateur proximal de la régularisation g par un débruiteur générique.

2.1 Un débruiteur "descente de gradient" ...

Nous proposons d'utiliser PnP-HQS avec un opérateur de débruitage D_σ appelé GS (pour *gradient step*) qui prend la forme d'une étape de descente de gradient

$$D_\sigma = \text{Id} - \nabla g_\sigma, \quad (3)$$

avec $g_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}$. Le choix de la paramétrisation de g_σ est fondamental pour les performances de débruitage. Nous avons observé expérimentalement que paramétrer directement g_σ par un réseau de neurones ne permet pas de débruiter efficacement. Afin d'utiliser la capacité de représentation des meilleurs débruiteurs profonds, nous proposons de définir

$$g_\sigma(x) = \frac{1}{2} \|x - N_\sigma(x)\|^2 \quad (4)$$

où $N_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est paramétré par un réseau de neurones de classe \mathcal{C}^1 . En notant $J_{N_\sigma}(x)$ la jacobienne de N_σ au point x , ceci mène à

$$\begin{aligned} D_\sigma(x) &= x - \nabla g_\sigma(x) \\ &= N_\sigma(x) + J_{N_\sigma}(x)^T (x - N_\sigma(x)) \end{aligned} \quad (5)$$

Grâce à cette écriture, nous pouvons paramétrer N_σ avec n'importe quelle architecture de réseau de neurones différentiable qui s'est montrée efficace pour le débruitage d'image. Malgré le fait que la capacité de représentation du débruiteur est limitée par la forme particulière (5), nous verrons en Section 4.1 que cette paramétrisation permet encore d'obtenir des résultats de débruitage à l'état de l'art.

2.2 ... Pour une minimisation explicite

En utilisant le débruiteur "GS" (3), l'opérateur PnP-HQS se réécrit $T_{\text{PnP-HQS}} = D_\sigma \circ \text{Prox}_{\tau f} = (\text{Id} - \nabla g_\sigma) \circ \text{Prox}_{\tau f}$. Afin de faire correspondre cet algorithme avec celui de descente de gradient proximal (PGD), nous proposons d'échanger l'ordre des opérations sur f et g_σ et de relaxer le débruitage avec un paramètre $\lambda \geq 0$. Notre algorithme GS-PnP s'écrit $x_{k+1} = T_{\text{GS-PnP}}^{\tau, \lambda}(x_k)$ avec

$$\begin{aligned} T_{\text{GS-PnP}}^{\tau, \lambda} &= \text{Prox}_{\tau f} \circ (\tau \lambda D_\sigma + (1 - \tau \lambda) \text{Id}), \\ &= \text{Prox}_{\tau f} \circ (\text{Id} - \tau \lambda \nabla g_\sigma). \end{aligned} \quad (6)$$

Avec de bonnes hypothèses sur f et g_σ (voir Section 3), les points fixes de l'opérateur PGD $T_{\text{GS-PnP}}^{\tau, \lambda}$ correspondent à des points critiques de la fonctionnelle

$$F(x) = f(x) + \lambda g_\sigma(x). \quad (7)$$

Ainsi, utiliser le débruiteur GS de l'équation (3) revient à inclure un terme de régularisation explicite et mène à un problème d'optimisation explicite résolu par l'algorithme PnP.

3 Analyse de la convergence

Dans cette section, nous introduisons les conditions sur f et D_σ qui permettent la convergence des itérations PnP (6) vers une solution de (7). Pour cela, nous exploitons des résultats sur la convergence de l'algorithme PGD en non-convexe [2, 3].

En restauration d'image, il est usuel de choisir un modèle de bruit gaussien, ce qui mène à une attache aux données convexe et lisse de la forme $f(x) = \|Ax - y\|^2$. Afin de gérer d'autres modèles de bruits, comme Laplace ou Poisson, nous supposons seulement que f est propre, semi-continue inférieurement et convexe. Ensuite, la régularisation g_σ est supposée différentiable avec un gradient Lipschitz, mais pas nécessairement convexe. Ces hypothèses sont compatibles avec la paramétrisation de g_σ présentée en Section 2. Sans plus d'hypothèses sur f et g_σ , le Théorème 1 établit la convergence des valeurs de la fonctionnelle et du résidu. Nous obtenons de plus la convergence des itérés en supposant que $F = f + \lambda g_\sigma$ est coercive et vérifie la propriété de Kurdyka-Lojasiewicz (KL) [1]. La coercivité de F est obtenue en ajoutant à g_σ un terme de rappel à un ensemble convexe compact (voir [6, Appendix D]). La propriété KL est vraie pour une très large classe de fonctions et est en pratique toujours vérifiée pour les problèmes inverses étudiés.

Théorème 1 (Preuve dans [6]). *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ et $g_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}$ des fonctions propres, semi-continues inférieurement avec f convexe et g_σ différentiable à gradient L -Lipschitz. Soit $\lambda > 0$, $F = f + \lambda g_\sigma$ et supposons F bornée inférieurement. Alors, pour $\tau < \frac{1}{\lambda L}$, les itérés x_k donnés par le schéma itératif (6) vérifient*

- (i) $(F(x_k))$ est décroissant et converge.
- (ii) Le résidu $\|x_{k+1} - x_k\|$ converge vers 0.
- (iii) Tous les points d'accumulation de la suite (x_k) sont des points stationnaires de (7).

Si F est coercive et vérifie la propriété KL, alors (x_k) converge, en un nombre fini d'itérations, vers un point stationnaire x^* de F .

4 Expériences

Nous analysons à présent les performances du débruiteur GS, puis confirmons empiriquement la convergence de la méthode GS-PnP et sa capacité à restaurer efficacement des images.

4.1 Débruiteur GS

Architecture du débruiteur Nous choisissons de paramétrer N_σ avec l'architecture DRUNet [19]), un réseau U-Net dans lequel des blocs résiduels sont intégrés. DRUNet prend le niveau de bruit σ en entrée, ce qui est cohérent avec la formulation proposée. Afin d'assurer la différentiabilité de N_σ par rapport à l'entrée, nous remplaçons les activations RELU par des activations ELU. Pour limiter le temps d'inférence du débruiteur, nous limitons à 2 le nombre de blocs résiduels par échelle.

Conditions d’entraînement Nous utilisons la base de données d’entraînement d’images couleurs proposée par [19]. Pendant l’entraînement, les images sont bruitées avec un bruit blanc gaussien ξ_σ d’écart type σ choisi au hasard dans l’intervalle $[0, 50/255]$. Avec notre paramétrisation (5) de D_σ , le réseau N_σ est entraîné pour minimiser le coût de reconstruction L^2 . Pour chaque batch, le gradient ∇g_σ est calculé grâce aux outils de différentiation automatique de PyTorch.

Résultats de débruitage Nous évaluons les performances PSNR du débiteur GS (GS-DRUNet) sur des images couleurs de taille 256×256 de la base de données CBS68 [11]. Nous comparons dans le Tableau 1, pour différents niveaux de bruit σ , notre modèle avec le débiteur DRUNet qui a la même architecture que GS-DRUNet mais qui n’est pas contraint à être un champ de vecteurs conservatif. Nous comparons aussi GS-DRUNet avec les deux débruiteurs profonds de référence FFDNet [22] et DnCNN [20]. Nous indiquons enfin dans le Tableau 1 le temps moyen requis, par chaque modèle, pour traiter une image 256×256 sur un GPU Tesla P100.

| σ (./255) | 5 | 15 | 25 | 50 | Time (ms) |
|------------------|--------------|--------------|--------------|--------------|-----------|
| FFDNet | 39.95 | 33.53 | 30.84 | 27.54 | 1.9 |
| DnCNN | 39.80 | 33.55 | 30.87 | 27.52 | 2.3 |
| DRUNet | <u>40.19</u> | <u>33.89</u> | <u>31.25</u> | <u>28.00</u> | 6.3 |
| GS-DRUNet | 40.26 | 33.90 | 31.26 | 28.01 | 10.4 |

TAB. 1: PSNR(dB) et temps d’inférence moyens lors du débruitage des images couleurs de CBS68, pour différents niveaux de bruit σ .

Bien que contraint à être un champ de vecteurs conservatif, notre débiteur GS-DRUNet atteint, et même dépasse légèrement, les performances du débiteur DRUNet non contraint. La différence de temps d’inférence entre ces deux modèles, due au calcul du gradient, reste raisonnable. Ces résultats montrent que le débiteur GS-DRUNet est un bon candidat pour être utilisé dans un algorithme Plug-and-Play.

4.2 Défloutage Plug-and-Play

Nous montrons dans cette section que l’algorithme GS-PnP proposé converge et atteint des performances au niveau de l’état de l’art pour le défloutage d’images. Des expériences de super-résolution ou d’inpainting sont également proposées dans [6].

Nous cherchons une estimation x de l’image propre $\hat{x} \in \mathbb{R}^n$ à partir d’une observation $y = H\hat{x} + \xi_\nu$ où H est un opérateur de convolution réalisé avec des conditions de bord circulaires et où ξ_ν est un bruit blanc gaussien d’écart type ν . La fonctionnelle minimisée par GS-PnP est $F = f + \lambda_\nu g_\sigma$ avec le terme d’attache aux données $f(x) = \frac{1}{2} \|Hx - y\|^2$ et $\lambda_\nu = \lambda \nu^2$ (multiplication of the objective function by ν^2). Comme la constante de Lipschitz de ∇g_σ est inconnue, nous rajoutons à chaque itération une étape de "backtracking" [3, Chapter 10] qui permet d’adapter le pas de temps pour assurer la décroissance de la fonctionnelle. Avec cette formulation, la convergence des itérés et des valeurs de la fonctionnelle est garantie

par le Théorème 1. Nous proposons un choix de paramètres de régularisation ($\lambda_\nu = 0.1$ et $\sigma = 0.8\nu$) qui permet une restauration efficace sur un ensemble de noyaux de flous et de niveaux de bruit. L’influence de ces paramètres est détaillée dans [6].

Nous montrons l’efficacité de notre méthode sur un ensemble de 10 noyaux de flou (8 flous de bougé de [8], un flou uniforme et un flou gaussien) et 3 niveaux de bruit gaussien $\nu \in \{2.55, 7.65, 12.75\}/255$.

Nous comparons dans le Tableau 2 notre méthode (GS-PnP) à la méthode par patch EPLL [23] et aux algorithmes PnP avec les débruiteurs paramétrés par des réseaux de neurones IRCNN [21], DPIR [19], et RED [13]. Ces résultats montrent que GS-PnP atteint les performances de DPIR, à l’état de l’art pour le défloutage par méthode Plug-and-Play. Nous soulignons que la convergence de GSPnP est garantie alors que DPIR peut diverger asymptotiquement, comme montré dans [6].

| ν (./255) | 2.75 | 7.65 | 12.75 |
|---------------|--------------|--------------|--------------|
| EPLL | 27.34 | 25.16 | 24.04 |
| RED | 29.17 | 25.26 | 22.84 |
| IRCNN | 31.97 | 28.58 | 26.94 |
| DPIR | 32.50 | 28.91 | <u>27.42</u> |
| GS-PnP | <u>32.33</u> | <u>28.90</u> | 27.44 |

TAB. 2: PSNR(dB) moyen lors du défloutage des images couleur sur la base de données CBS10 (sous-ensemble de CBS68) avec différents noyaux de flou et différents niveaux de bruit ν .

Nous comparons visuellement Figure 1(c-f) le défloutage obtenu avec les différentes méthodes sur une même image. On observe que notre algorithme restaure les structures les plus nettes. Nous représentons aussi les courbes de convergence qui confirment empiriquement la convergence des valeurs de la fonctionnelle et du résidu.

5 Conclusion

Dans ce travail, nous introduisons un nouvel algorithme PnP avec des garanties théoriques de convergence. Un débiteur est entraîné pour réaliser une étape de descente de gradient sur une fonction de régularisation qui est paramétrée par un réseau de neurones. Ce débiteur est utilisé dans un schéma itératif basé sur la méthode PnP-HQS. Il est démontré que l’algorithme converge vers un point stationnaire d’une fonctionnelle explicite. Notre approche permet de traiter des termes d’attaches aux données non fortement convexes avec un débiteur qui n’est pas non-expansif. Les expériences de défloutage réalisées confirment les résultats de convergence théoriques et montrent que la méthode atteint des performances au niveau de l’état de l’art.

References

- [1] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for

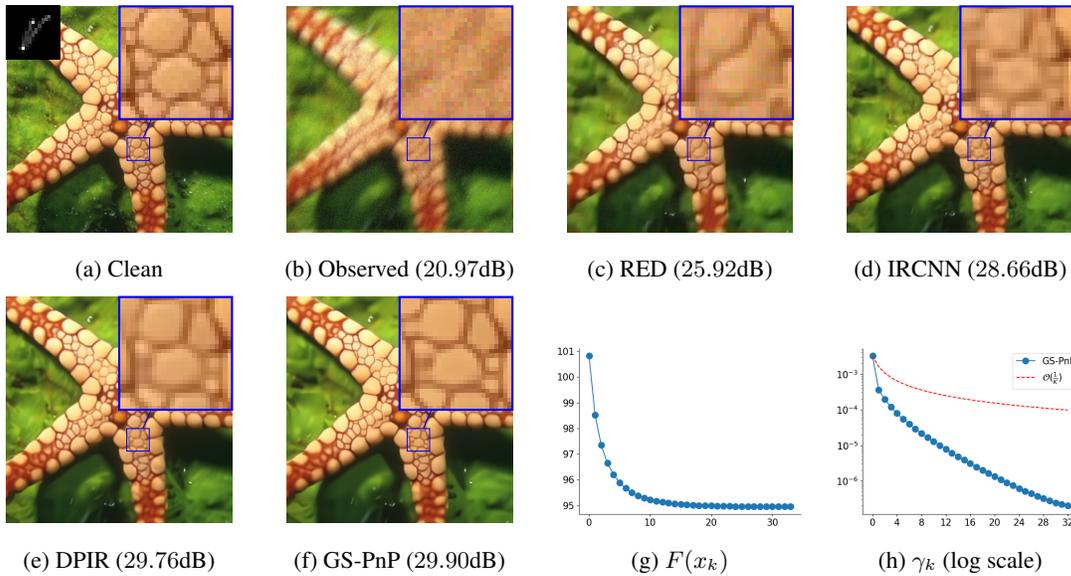


FIG. 1: Défloutage avec différentes méthodes de l'image "starfish" dégradée avec le noyau de flou indiqué et un bruit de niveau $\nu = 0.03$. En (g) and (h), nous montrons l'évolution de $F(x_k)$ et $\gamma_k = \min_{0 \leq i \leq k} \|x_{i+1} - x_i\|^2 / \|x_0\|^2$.

- nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Math. of Oper. Res.*, 2010.
- [2] H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods. *Math. Program.*, 2013.
- [3] A. Beck. *First-order methods in optimization*. 2017.
- [4] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. 2011.
- [5] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE TIP*, 1995.
- [6] S. Hurault, A. Leclaire, and N. Papadakis. Gradient step denoiser for convergent plug-and-play. In *ICLR*, 2022.
- [7] S. Hurault, A. Leclaire, and N. Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. *arXiv preprint arXiv:2201.13256*, 2022.
- [8] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE CVPR*, 2009.
- [9] S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems. In *NeurIPS*, 2018.
- [10] S. Mallat. *A Wavelet Tour of Signal Processing, The Sparse Way*. 2009.
- [11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE ICCV*, 2001.
- [12] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la SMF*, 1965.
- [13] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (RED). *SIAM SIIMS*, 2017.
- [14] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 1992.
- [15] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin. Plug-and-play methods provably converge with properly trained denoisers. In *ICML*, pages 5546–5557, 2019.
- [16] M. Terris, A. Repetti, J.-C. Pesquet, and Y. Wiaux. Building firmly nonexpansive convolutional neural networks. In *IEEE ICASSP*, pages 8658–8662, 2020.
- [17] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE GlobalSIP*, 2013.
- [18] X. Yuan, Y. Liu, J. Suo, and Q. Dai. Plug-and-play algorithms for large-scale snapshot compressive imaging. In *IEEE CVPR*, 2020.
- [19] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE PAMI*, 2021.
- [20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, 2017.
- [21] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR*, 2017.
- [22] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE TIP*, 2018.
- [23] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE ICCV*, 2011.